BRINGING DEEP LEARNING INTO BIG DATA ANALYTICS USING BIGDL

Xianyan Jia (xianyan.jia@intel.com) Big Data Technology, Software and Service Group, Intel
Zhenhua Wang (metas.wzh@gmail.com) AI and big data, JD.com
What Is BigDL?

- Distributed deep learning framework for Apache Spark*.

- Feature parity with popular deep learning frameworks
  - E.g., Caffe, Torch, Tensorflow, etc.

- High performance
  - Powered by Intel MKL and multi-threaded programming

- Efficient scale-out
  - Leveraging Spark for distributed training & inference

Github: github.com/intel-analytics/BigDL
http://software.intel.com/bigdl
Run as standard Spark Programs

Goal: Make deep learning more accessible to big data users and data scientists

• **No changes** to the Spark or Hadoop clusters needed

Iterative

• Each iteration of the training/inference runs as a Spark job

Data parallel

• Each Spark task runs the same model on a subset of the data (batch)
Why BigDL?

Production ML/DL system is **Complex and Distributed.**

Spark-based Deep Learning library is a natural fit

---

Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

"Hidden Technical Debt in Machine Learning Systems",
Google, NIPS 2015 Paper

https://github.com/intel-analytics/BigDL
Why BigDL

BigDL: Run deep learning on Big Data platform

Outstanding features

- Massively distributed
- Fault tolerance
- Elasticity
- Dynamic resource sharing
- ...

https://github.com/intel-analytics/BigDL
Build a distributed image prediction pipeline on Spark using BigDL

https://github.com/intel-analytics/BigDL
Read from Distributed Cluster

- Read Folder
- Read Apache Parquet
- RDD

ImageFrame

https://github.com/intel-analytics/BigDL
Vision Library on top of OpenCV

Pascal VOC data sets (http://host.robots.ox.ac.uk/pascal/VOC/)
Data Loading and Pre-processing Pipeline

```
// read distributed images and perform preprocessing
val distImageFrame = DataFrame.read(folder, sc) ->
  BytesToMat() -> Resize(300, 300) ->
  MatToFloats(meanRGB = Some(123, 117, 104))
```

https://github.com/intel-analytics/BigDL
Model Zoo

Image Classification
- Inception
- Resnet
- VGG
- MobileNet
- Alexnet
- DenseNet
- SqueezeNet

Object Detection
- SSD (Single Shot Multibox Detector)
  - VGG
  - MobileNet
- Faster-RCNN
  - VGG
  - PvaNet

https://github.com/intel-analytics/BigDL
Image Recognition and Object Detection

Pascal VOC data sets (http://host.robots.ox.ac.uk/pascal/VOC/)

https://github.com/intel-analytics/BigDL
Object Detection Video Demo

https://github.com/intel-analytics/analytics-zoo/blob/master/apps/ssd/SSD.ipynb
Load model

TensorFlow → Caffe → BigDL → torch

https://github.com/intel-analytics/BigDL
Load model

// load BigDL model
val model = Module.loadModule(path)
// load caffe model
val modelCaffe = Module.loadCaffeModel(caffeDefPath, caffeModelPath)
// load tensorflow model
val modelTF = Module.loadTF(graphFile, inputs, outputs, byteOrder, binFile)
// load torch model
val modelTorch = Module.loadTorch(path)
Build a distributed image prediction pipeline on Spark using BigDL

```scala
val distImageFrame = ImageFrame.read(folder, sc) -> preprocessor
val model = Module.loadModule(path)
model.predict(distImageFrame)
distImageFrame.save(outPath)
```
Problem

Large-scale image feature extraction

• Object detect (remove background, optional)
• Feature extraction

Application

• Similar image search
• Image Deduplication
Similar image search
Image Deduplication

HDFS

Feature data:
1, image_id
2, feature vector

SPARK
1, sampling & modeling
2, partitioning based on the mode according to similarity
3, do step 1 and 2 recursively until the size of leaf node is below the threshold bound
4, density based partitioning

Algorithm library
libcluster.so
Feature Extraction pipeline

https://mp.weixin.qq.com/s/xUckzbHK4K06-v5qUsaNOQ (Chinese)
https://software.intel.com/en-us/articles/building-large-scale-image-feature-extraction-with-bigdl-at-jdcom (English)

Challenges of Large-Scale Processing in GPU Solutions

• Deploy deep learning inference application in GPU or FPGA is very expensive
  • GPU: The total count GPUs are much less than CPUs in our case. The most of GPUs are used to train models.
  • FPGA: requires a long life cycle to compile, test and deploy deep model to FPGA.

• Very complex and error-prone in managing large-scale GPU solutions in non-cluster mode
  • E.g., resource management and allocation, data partitioning, task balance, fault tolerance, model deployment, etc.

• Low parallelism in GPU solutions in cluster mode
  • GPU cores: 1 executor-cores and at most 4 num-executors are permitted due to limit of cards in case of 4 per server when using Caffe lib.
  • GPU memory: is very expensive and size is very limited.

http://mp.weixin.qq.com/s/xUckzbHK4K06-v5qUsaNOQ (Chinese)
https://software.intel.com/en-us/articles/building-large-scale-image-feature-extraction-with-bigdl-at-jdcom (English)
Upgrade to BigDL + Xeon Solutions

- Reuse existing Hadoop/Spark clusters for deep learning with no changes
- Efficiently scale out on Spark with superior performance
  - 3.83x speed-up when running on ~24 Xeon servers vs. 20 K40 GPU cards
- Very easy to build the end-to-end pipeline in BigDL
  - Library of image transformation and augmentation based on OpenCV
    
    ```scala
    val preProcessor = BytesToMat() -> Resize(300, 300) -> ... 
    val transformed = preProcessor(dataRdd)
    ```
  - Loading pre-trained model (Caffe / Torch / TensorFLow)
    
    ```scala
    val model = SSDCaffeLoader.loadCaffe(
      caffeDefPath, caffeModelPath)
    ```
Pipeline Correctness

Almost same as Caffe GPU
Element-wise error < 0.001%
3.83x Speed up compared to GPU solution
Lessons

- OpenCV memory release
  - Be careful with operations with new OpenCV memory, e.g. split
- Deal with corrupted images
  - Mark them as invalid instead of job crash
Blogs

- In English: https://software.intel.com/en-us/articles/building-large-scale-image-feature-extraction-with-bigdl-at-jdcom
- In Chinese: http://mp.weixin.qq.com/s/xUCkzbHK4K06-v5qUsaNQQ
Reference

• BigDL, https://github.com/intel-analytics/BigDL

• Liu, Wei, et al., SSD: Single Shot MultiBox Detector, European conference on computer vision. Springer, Cham, 2016.


• Open Source Computer Vision Library, http://opencv.org/
Thanks
Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A “Mission Critical Application” is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined”. Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Intel, Quark, VTune, Xeon, Cilk, Atom, Look Inside and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright ©2017 Intel Corporation.
Risk Factors

The above statements and any others in this document that refer to plans and expectations for the first quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as “anticipates,” “expects,” “intends,” “plans,” “believes,” “seeks,” “estimates,” “may,” “will,” “should” and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the company's expectations. Demand could be different from Intel's expectations due to factors including changes in business and economic conditions; customer acceptance of Intel's and competitors' products; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Uncertainty in global economic and financial conditions poses a risk that consumers and businesses may defer purchases in response to negative financial events, which could negatively affect product demand and other related matters. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Revenue and the gross margin percentage are affected by the timing of Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel's response to such actions; and Intel's ability to respond quickly to technological developments and to incorporate new features into its products. The gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; start-up costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; product manufacturing quality/yields; and impairments of long-lived assets, including manufacturing, assembly/test and intangible assets. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. Intel's results could be affected by the timing of closing of acquisitions and divestitures. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel's ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. A detailed discussion of these and other factors that could affect Intel's results is included in Intel's SEC filings, including the company's most recent reports on Form 10-Q, Form 10-K and earnings release.